A Landscape of Interval Life-like Freezing Cellular Automata

Eric Goles, Diego Maldonado, Pedro Montealegre, Martín Ríos-Wilson

A Landscape of Interval Life-like Freezing Cellular Automata

Eric Goles*[1], Diego Maldonado†[1], Pedro Montealegre‡[2], and Martín Ríos-Wilson§[1]

[1]Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Santiago, Chile
[2]Departamento de Ingeniería informática, Facultad de Ingeniería, Universidad Católica de la Santísima Concepción, Concepción, Chile

Abstract

A two-state Cellular Automaton (CA) with neighborhood N is called an *Interval Life-Like Freezing Cellular* if the local rule f can be defined from an interval $\mathcal{I} \subset [0, \ldots, |N|]$, such that: (1) when a cell c is in state 0, it becomes 1 if and only if the number of neighbors of c in state 1 belongs to $\mathcal{I}$. (2) if c is in 1, then it remains in state 1 independently on the states of their neighbors.

Recently (Goles et al. Information and Computation 2020), an exhaustive study of the dynamics and complexity of ILLFCA was carried out in the von Neumann neighborhood, where the complexity and dynamics of such CAs were classified in five groups, namely *Trivial, Algebraic, Topological, Fractal* and *P-Complete* rules.

In this exploratory paper, we propose a follow-up of the later research, extending it to ILLFCA equipped with the Moore neighborhood. Interestingly, this class contains the famous *Life-without-Death*, also known as the freezing version of *Conway Game of Life*. In fact, we show that the family of ILLFCA with Moore neighborhood has a rich diversity of rules, some of them not classifiable in the previous groups.

*eric.chacc@uai.cl
†dmaldonado@ucsc.cl
‡p.montealegre@uai.cl
§martin.rios@edu.uai.cl

1 Introduction

In this exploratory paper, our goal is to explore an exhaustive classification of the complexity of the Interval Life-Like Freezing Cellular Automata with Moore neighborhood. We begin defining this class.

A *freezing cellular automaton* is a CA that admits an order on its states, such that the state evolution of any node is non-decreasing in any orbit. Several models that received a lot of attention in the literature are actually freezing dynamics, for instance: self-assembly tilings [1], forest fire or epidemic [2, 3] propagation models, bootstrap percolation [4], etc. On the other hand, their complexity as computational models has been studied from various points of view: as language recognizers where they correspond to bounded change or bounded communication models [5], for their computational universality [6, 7], as well as for various associated decision problems [8].

A *Life-Like Cellular Automata* are a family of CAs that generalize *Conway's Game of Life* [9]. This family contains the totalistic CA with state $\{0,1\}$ and Moore neighborhood N such that there exist sets $B, S \subseteq \{0, 1..., 8\}$ such that:

$$f(x_N) = 1 \Leftrightarrow \underbrace{(x_{(0,0)} = 0 \wedge \sum_{z\in N} x_i \in B)}_{\text{Born rule}} \vee \underbrace{(x_{(0,0)} = 1 \wedge \sum_{z\in N} x_i \in S)}_{\text{Survivor rule}}$$

A Life-Like CA is called *Interval Life-Like Cellular Automaton* (ILLFCA) if both sets B and S are closed intervals. For instance, the Game of Life is the Interval Life-Like CA with $B = \{3\}$, $S = \{2, 3\}$. Observe that an Interval Life-Like CA is *freezing* if $S = \{0, 1..., 8\}$. The *freezing version* of the Game of Life is known as *the Life Without Death* [10] ($B = \{3\}$, $S = \{0, 1, ..., 8\}$).

Since we only consider Interval Life-Like CA that are freezing, we omit the set S in their definition, as these rules are only defined by the interval B. Moreover, for $0 \leq a \leq b \leq 8$, we denote $R_{a,b}$ the ILLFCA such that $B = [a, b]$. For instance the Life Without Death is denoted $R_{3,3}$. In order to illustrate the different rules that belong to this family, we present them in the table shown in Figure 1. In the i-th row of the latter table we put all the ILLFCAs whose interval starts at i and in column j all those ILLFCAs whose interval closes at j. Thus, we obtain an upper triangular matrix (see Figure 1). From here we identify four types of rules:

- The rules $R_{i,8}$ for $i \in \{1, \ldots, 7\}$, called *threshold rules*. In these rules a cell switches to state 1 if at least a threshold i of its neighbors are in state 1. These rules are denoted by θ_i.

- The rules $R_{1,j}$ for $j \in \{2, \ldots, 7\}$, called *anti-threshold rules*. In these rules a cell switches to state 1 if the set of neighbors in state 1 is at most i. These rules are denoted $\overline{\theta}_i$.

- Finally, the rules $R_{i,i}$ for $i \in \{1, \ldots 8\}$, where a cell is born if there are exactly i living neighbors are called *delta rules.* These rules are denoted δ_i.

- The *inner rules* are all the remaining rules, that is to say, all rules $R_{i,j}$ such that $1 < i < j < 8$.

δ_1	$\overline{\theta}_2$	$\overline{\theta}_3$	$\overline{\theta}_4$	$\overline{\theta}_5$	$\overline{\theta}_6$	$\overline{\theta}_7$	θ_1
	δ_2	$R_{2,3}$	$R_{2,4}$	$R_{2,5}$	$R_{2,6}$	$R_{2,7}$	θ_2
		δ_3	$R_{3,4}$	$R_{3,5}$	$R_{3,6}$	$R_{3,7}$	θ_3
			δ_4	$R_{4,5}$	$R_{4,6}$	$R_{4,7}$	θ_4
				δ_5	$R_{5,6}$	$R_{5,7}$	θ_5
					δ_6	$R_{6,7}$	θ_6
						δ_7	θ_7
							δ_8

Figure 1: Family of ILLFCAs. In blue we highlight the threshold rules, in gray the anti-threshold rules, and in red the delta rules. All the remaining are inner rules.

We remark that we do not study the rules $R_{0,b}$. This is because all these rules satisfy the following property: after the first iteration, all cells with 0 alive neighbors become alive. Then, we can consider that in the next iterations of the dynamics the rule $R_{1,b}$ is applied. Thus, studying the $R_{0,b}$ rules is equivalent to study the $R_{1,b}$ rules with initial configurations without isolated neighbors.

In order to measure the complexity of each rule, we consider the computational complexity of the decision problem consisting in predicting the future state of each cell. More precisely, given an initial condition and a cell in state 0 we would like to predict if it will eventually change to state 1 or it will be stay fixed in state 0. Observe that since the rules we are considering are freezing, this latter task corresponds to study of the fixed points [11]. We say that a cell is *stable*, if it is initially not alive and never changes. Thus, when the initial configuration reaches a fixed point, the remaining non-living cells are the stable cells. Conversely, the alive cells at the fixed point are the non-stable (unstable) cells or they are cells that were initially in state 1. The definition of stability introduces the following decision problem:

> Stability$_{R_{a,b}}$
> **Input:** a periodic configuration x described by $\{0,1\}^{[n]\times[n]}$ and a site $u \in [n] \times [n]$.
> **Question:** Is u a stable cell for x?

The computational complexity of a decision problem can be defined as the amount of resources (i.e. time and space) required to solve it. Observe that, since the rules are freezing, a fixed point is always reached in polynomial time in the size of the initial condition. Indeed, in every time-step before reaching an attractor at least one cell is frozen in state 1. Thus, there is always a (deterministic) polynomial-time algorithm for solving Stability, which consists in simulating the rule $R_{a,b}$. Considering this latter observation (and from a classical computational complexity standpoint) we can roughly identify the following well-known time-complexity classes: **P**, the class of problems solvable in polynomial time on a sequential computer, and **NC**, the class of problems solvable in poly-logarithmic time in a PRAM machine, with a polynomial number of processors [12]. Roughly speaking, **NC** is the class of problems for which there exists a *fast parallel algorithm* which solves them. Observe that, since any parallel algorithm can be straightforwardly executed sequentially (the execution times depends on the number of processors and the time that each processor takes in the parallel computation), we have that $\mathbf{NC} \subseteq \mathbf{P}$. In this context, it is well-believed that $\mathbf{NC} \neq \mathbf{P}$, and so, there exist "inherently sequential" problems, i.e. problems that are in **P** but they are not in **NC**. Those inherently sequential problems are usually called **P**-complete problems. More precisely, a decision problem is **P**-complete if it is in **P** and if any other problem in **P** is reducible (by an **NC**-reduction or a logarithmic space reduction) to that decision problem. Thus, if for some **P**-complete problem, there is a fast parallel algorithm solving it, then necessarily **P**=**NC** [12, 13].

Griffeath and Moore [14] showed that Stability is **P**-complete for the Life Without Death. This was accomplished by simulating arbitrary boolean circuits by a series of gadgets representing logic gates as configuration patterns. This was similar to the work done by Banks [15]. On the other hand, Goles et al. [8] showed that on a regular graph of degree n, the stability problem for a threshold rule with threshold $n-1$ is in **NC**. Thus, for θ_7, Stability is in **NC**.

Considering the von Neumann neighborhood Goles et al. [16] studied Stability for two-state freezing totalistic automata. They classified the latter rules in five groups:

Trivial: It is easy to determine whether a cell will change. Dynamics are of the following types: any initial pattern fills everything, initial configurations remain unchanged, or a fixed point is reached in a few iterations.

Algebraic: The rule, or part of it, can be computed as a composition of associative operation, which is used to quickly compute the fixed point of any initial configuration. Dynamics are waves of live cells that collide and, depending on the position of the cells at the point of collision, may or may not change.

Topological: The set of stable cells can be characterized according to the connectedness of the induced graph in the grid. In the case of the strict majority, the characterization is that the stable cells are initially inactive cells that form biconnected components or paths that connect them. In the case of the non-strict majority, the stable cells are those that form a 3-connected component of initially inactive cells.

P-complete: are rules capable of simulating logic circuits. This is the only case where Stability was shown to be **P**-complete.

Fractal: The dynamics are of patterns that grow in a self-repeating way. For this set, the complexity of Stability was not settled.

2 A landscape of Interval Life-like Freezing CA

By analyzing the rules and performing numerical experiments, we observe that the ILLFCA with the Moore neighborhood can be grouped in a similar way than it was done in [16] for the von Neumann case. Nevertheless, we find a new set of rules, that we denote *signal rules*, which exhibit a behavior not found in the von Neumann case. We now detail the results regarding the threshold, anti-threshold and delta rules.

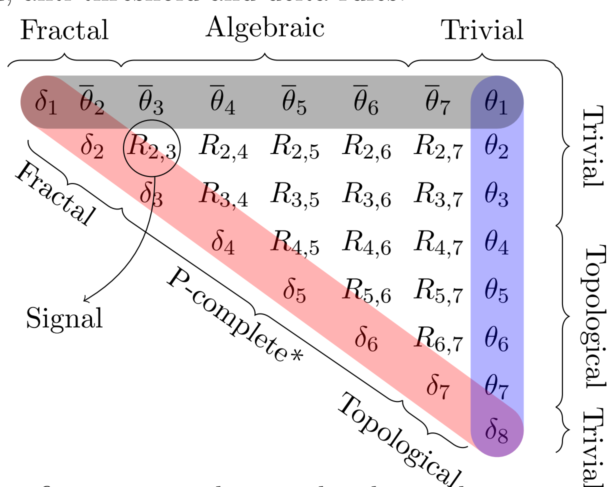


Figure 2: Classification according to the observed in computational simulations. For those marked with *analytical complexity proofs are provided.

Trivial rules: $R_{1,8}, R_{2,8}, R_{3,8}, R_{1,7}, R_{7,7}$ and $R_{8,8}$. It is easy to see that for rule $R_{1,8}$ the two possible fixed points are all alive or all dead configurations. Rule $R_{1,7}$ is similar, but admits some fixed points with dead cells having all neighbors alive. Rule $R_{8,8}$ is trivial as every configuration reaches a fixed point in one time-step. For rule $R_{2,8}$ we have that a configuration has at least two alive cells sharing the same neighborhood, then everything converges to the configurations where all the cells are alive. Finally, the case of the rule

$R_{3,8}$, is slightly more complicated as there are three-cell configurations that are stable (see Figure 4), but it is enough to study all patterns of a given fixed size. To the following figures, white means death and black means alive.

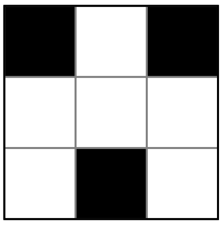


Figure 3: Pattern that forces any configuration to reach the all alive fixed point for rule $R_{3,8}$.

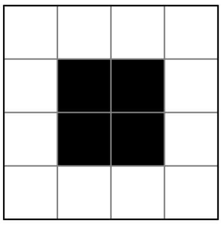


Figure 4: Stable pattern for rule $R_{3,8}$.

Fractal rules: $R_{1,1}, R_{1,2}, R_{2,2}$. In Figures 5, 6 we show examples of fractal dynamical behavior exhibited by rules $R_{1,1}$ and $R_{1,2}$ from a single alive cell. Rule $R_{2,2}$ requires more than one cell to exhibit such a behavior, since otherwise it does not change. In Figure 7 it is shown the behavior starting from three initial alive cells.

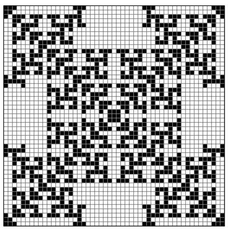


Figure 5: Rule $R_{1,1}$

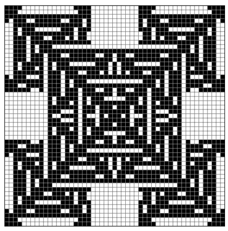


Figure 6: Rule $R_{1,2}$

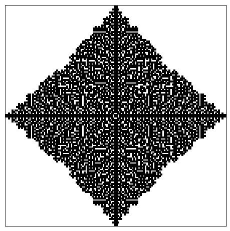
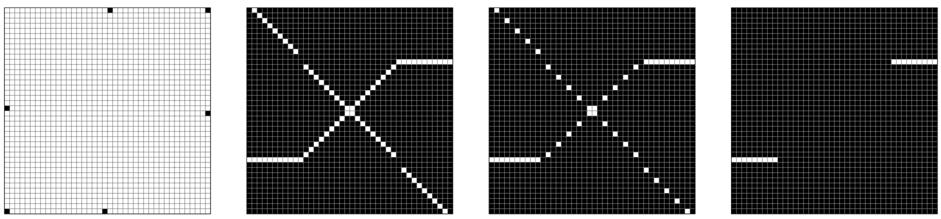


Figure 7: Rule $R_{2,2}$

Algebraic rules: $R_{1,3}, R_{1,4}, R_{1,5}, R_{1,6}$. In Figure 8 we show an example of the algebraic dynamic of all these rules. For all these rules, waves starting from alive cells propagate. Stable dead cells appear in the interface in which the different waves collide.



Initial configuration. Rule $R_{1,3}$. Rule $R_{1,4}$. Rule $R_{1,5}$.

Figure 8: Example of dynamics for the rules $R_{1,3}$, $R_{1,4}$, and $R_{1,5}$ starting from the same initial configuration.

Topological rules: rules $R_{4,8}, R_{5,8}, R_{6,8}, R_{7,8}, R_{7,7}$. We analyze each of these rules separately.

Rule $R_{4,8}$: This is the majority rule (also known as Bootstrap percolation). It can be shown that stable patterns are necessarily 4-connected. In addition, a sufficient condition for stability is 5-connectedness. Nevertheless, there are stable structures that are not 5-connected.

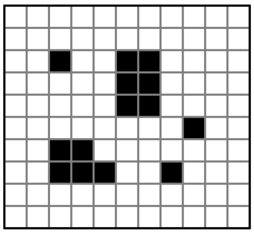


Figure 9: An example of a 5-connected pattern of stable cells.

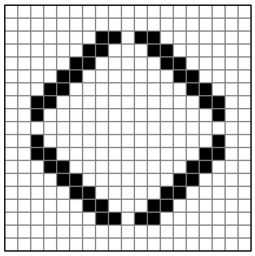


Figure 10: An example of a stable configuration which is not 5-connected.

Rule $R_{5,8}$: This is the freezing strict majority rule. Similar to the case of the von Neumann neighborhood, there exist stable bounded 4-connected components. Non-4-connected structures can be stabilized by attaching them to other 4-connected components (see Figures 11 and 12).

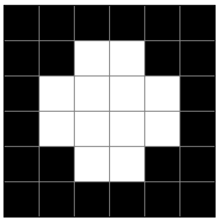


Figure 11: Stable bounded configuration.

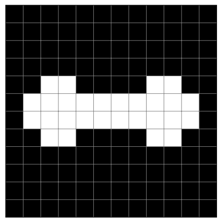


Figure 12: Two bounded configurations stabilizing a third non-4-connected one.

Rule $R_{6,8}$: In this case, there are bounded stable configurations. A sufficient condition for stability is 3-connectedness. However, there are a bounded non 3-connected components that are stable (see Figure 13).

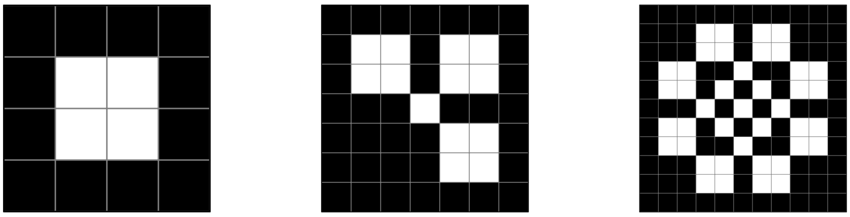


Figure 13: Left: Stable bounded configuration. Middle: Three Stable bounded configuration stabilizing a cell. Right Eight Stable patterns stabilizing a set of cells.

Rule $R_{7,8}$**:** As it is shown in [8], a cell is stable if and only if it is in a bi-connected component or in a path between them.

Rule $R_{7,7}$**:** This is almost a trivial rule. In order to decide if a cell is stable, it suffices to show that the cell belongs to a straight line of dead cells surrounded by alive cells (see Figure 14).

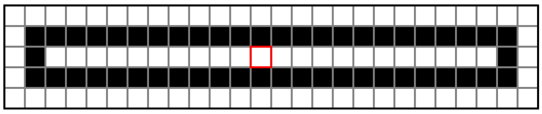


Figure 14: Example of a stable cell for rule $R_{7,7}$. Two signals arrive simultaneously to red cells, then the red pass from six to eight neighbors.

P-complete rules: $R_{3,3}, R_{4.4}, R_{5,5}$ and $R_{6,6}$. Rule $R_{3,3}$ is the Life Without Death, which is shown to be **P**-Complete in [14]. For the remaining rules, to show **P**-Completeness, we reduce the monotone circuit value problem [17] to Stability. To this end, we simulate an arbitrary monotone circuit by exhibiting gadgets which simulate the AND, OR, and XOR gates (see Figure 15) . The XOR gate will be used to simulate planar crossing (see e.g. [16]).

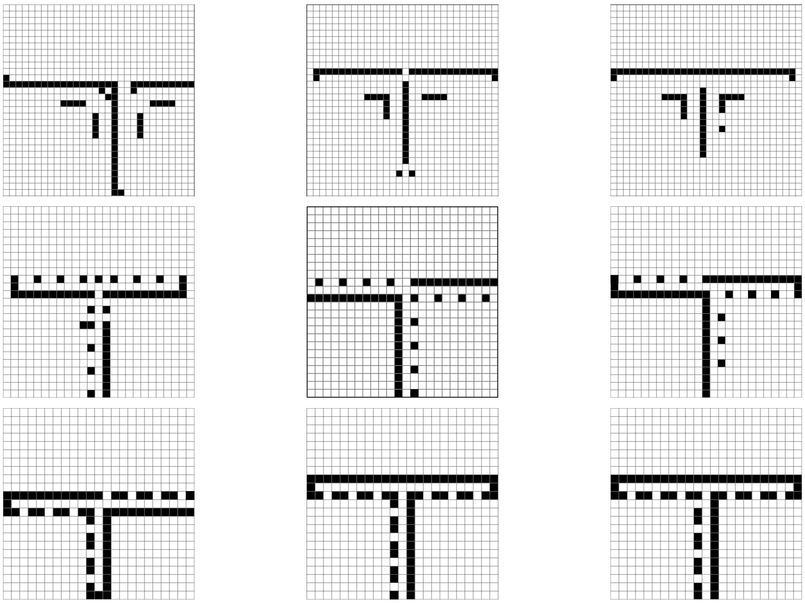


Figure 15: AND, OR and XOR gates (left, center, right) for rules $R_{4,4}, R_{5,5}$, and $R_{6,6}$ (top, middle, bottom), respectively

Signal rules: R_{23}. We see that the rule is capable of producing different types of signals (see Figures 16 and 17). This behavior could have two different implications: on the one hand, it could imply that the rule is capable of universal computation, as the case of rule 110 [18]. On the other hand, if the interactions of the signals are quite limited, it is possible that one can take advantage of that simplicity to develop an efficient algorithm for solving Stability. Interestingly, this is a type of behavior which was not observed in the von Neumann case.

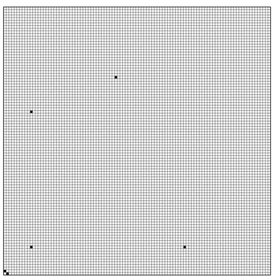


Figure 16: Rule $R_{2,3}$. Initial configuration.

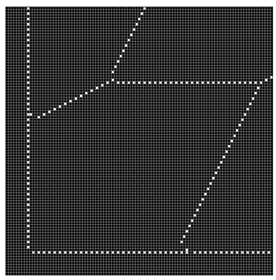


Figure 17: Rule $R_{2,3}$. Final configuration.

References

[1] Andrew Winslow. A brief tour of theoretical tile self-assembly. In *Cellular Automata and Discrete Complex Systems, AUTOMATA 2016*, pages 26–31, 2016.

[2] M.A. Fuentes and M.N. Kuperman. Cellular automata and epidemiological models with spatial dependence. *Physica A: Statistical Mechanics and its Applications*, 267(3–4):471 – 486, 1999.

[3] Per Bak, Kan Chen, and Chao Tang. A forest-fire model and some thoughts on turbulence. *Physics Letters A*, 147(5):297 – 300, 1990.

[4] Hamed Amini and Nikolaos Fountoulakis. Bootstrap percolation in power-law random graphs. *Journal of Statistical Physics*, 155(1):72–92, feb 2014.

[5] Martin Kutrib and Andreas Malcher. Cellular automata with sparse communication. *Theor. Comput. Sci.*, 411(38-39):3516–3526, 2010.

[6] E. Goles, N. Ollinger, and G. Theyssier. Introducing freezing cellular automata. In *Exploratory Papers of Cellular Automata and Discrete Complex Systems (AUTOMATA 2015)*, pages 65–73, 2015.

[7] Florent Becker, Diego Maldonado, Nicolas Ollinger, and Guillaume Theyssier. Universality in freezing cellular automata. In *CiE 2018*, pages 50–59, 2018.

[8] Eric Goles, Pedro Montealegre-Barba, and Ioan Todinca. The complexity of the bootstraping percolation and other problems. *Theoretical Computer Science*, 504:73–82, sep 2013.

[9] Martin Gardner. Mathematical games. *Scientific American*, 223(4):120–123, October 1970.

[10] Tommaso Toffoli and Norman Margolus. *Cellular Automata Machines: A New Environment for Modeling.* MIT Press, Cambridge, MA, USA, 1987.

[11] Eric Goles, Nicolas Ollinger, and Guillaume Theyssier. Introducing Freezing Cellular Automata. In *Cellular Automata and Discrete Complex Systems, 21st International Workshop (AUTOMATA 2015)*, volume 24 of *TUCS Lecture Notes*, pages 65–73, Turku, Finland, June 2015.

[12] Raymond Greenlaw, Howard Hoover, and Walter Ruzzo. *Limits to Parallel Computation: P-completeness Theory.* Oxford University Press, Inc., New York, NY, USA, 1995.

[13] Cristopher Moore. Majority-vote cellular automata, ising dynamics, and p-completeness. Working papers, Santa Fe Institute, 1996.

[14] David Griffeath and Cristopher Moore. Life without death is P-Complete. Working papers, Santa Fe Institute, 1997.

[15] Edwin Roger Banks. Universality in cellular automata. In *SWAT (FOCS)*, pages 194–215. IEEE Computer Society, 1970.

[16] Eric Goles, Diego Maldonado, Pedro Montealegre, and Nicolas Ollinger. On the complexity of the stability problem of binary freezing totalistic cellular automata. *Information and Computation*, 274:104535, 2020. AUTOMATA 2017.

[17] Leslie M. Goldschlager. The monotone and planar circuit value problems are log space complete for P. *SIGACT News*, 9(2):25–29, July 1977.

[18] Matthew Cook. Universality in elementary cellular automata. *Complex Systems*, 15(1):1–40, 2004.

Appendix

Definition 2.1. k-connectivity A connected graph is k-connected if and only if removing any $k-1$ vertices, the graph still connected.